mented. The computing device **398** may have a display **400**, a network interface **401**, as well as storage **402** and processing hardware **404**, which may be a combination of any one or more: central processing units, graphics processing units, analog-to-digital converters, bus chips, Field-programmable Gate Arrays (FPGAs), Application-specific Integrated Circuits (ASICs), Application-specific Standard Products (ASSPs), or Complex Programmable Logic Devices (CPLDs), etc. The storage **402** may be any combination of magnetic storage, static memory, volatile memory, etc. The meaning of the term "storage", as used herein does not refer to signals or energy per se, but rather refers to physical apparatuses, possibly virtualized, including physical media such as magnetic storage media, optical storage media, static memory devices, etc., but not signals per se. The hardware elements of the computing device **398** may cooperate in ways well understood in the art of computing. In addition, input devices **406** may be integrated with or in communication with the computing device **398**. The computing device **398** may have any form factor or may be used in any type of encompassing device. The computing device **398** may be in the form of a handheld device such as a smartphone, a tablet computer, a gaming device, a server, a rack-mounted or backplaned computer-on-a-board, a system-on-a-chip, or others.

[0062] Embodiments and features discussed above can be realized in the form of information stored in volatile or non-volatile computer or device readable media. This is deemed to include at least media such as optical storage (e.g., compact-disk read-only memory (CD-ROM)), magnetic media, flash read-only memory (ROM), or any current or future means of storing digital information. The stored information can be in the form of machine executable instructions (e.g., compiled executable binary code), source code, bytecode, or any other information that can be used to enable or configure computing devices to perform the various embodiments discussed above. This is also deemed to include at least volatile memory such as random-access memory (RAM) and/or virtual memory storing information such as central processing unit (CPU) instructions during execution of a program carrying out an embodiment, as well as non-volatile media storing information that allows a program or executable to be loaded and executed. The embodiments and features can be performed on any type of computing device, including portable devices, workstations, servers, mobile wireless devices, and so on.

1. A method performed by a computing device to interact with a storage device that comprises a controller and that is connected with the computing device via a bus of the computing device, the method comprising:

accessing a virtual disk file stored in a filesystem of the computing device, wherein content of the virtual disk file comprises virtual disk metadata and virtual blocks described by the virtual disk metadata;

parsing the virtual disk file to access the virtual disk metadata; and

instructing the storage device to form a virtual volume, the instructing including specifying device blocks for the virtual volume based on the parsed virtual disk metadata, wherein the virtual volume comprises an object managed by the storage device.

2. A method according to claim **1**, further comprising determining indirection mappings between the virtual blocks and device blocks, and specifying the device blocks according to the indirection mappings.

3. A method according to claim **1**, further comprising storing the content of the virtual disk file in the device blocks of the virtual volume.

4. A method according to claim **3**, wherein the storing comprises copying the content of the virtual disk file to the device blocks of the virtual volume.

5. A method according to claim **3**, wherein the storing comprises assigning device blocks that store the virtual disk file to the virtual volume.

6. A method according to claim **1**, wherein the storage device comprises a controller that implements a Non Volatile Memory express (NVMe) interface, wherein the virtual volume is stored as a NVMe namespace, and wherein the controller stores associations between device blocks and the NVMe namespace.

7. A method according to claim **1**, wherein the storage device stores virtual volume metadata in association with the virtual volume, the virtual volume metadata comprising structural data and/or indirection metadata, the structural data indicating at least a block size of the virtual volume, the indirection metadata mapping device blocks of the virtual volume to virtual blocks whose contents are stored in the mapped-to virtual blocks, the method further comprising requesting the virtual volume metadata from the storage device and using the indirection metadata and/or the structural metadata to automatically configure a second virtual disk file.

8. A computing device comprising:

processing hardware and a storage device, the storage device storing an operating system, the operating system comprising a filesystem module;

the storage device storing a filesystem volume configured to be managed by the filesystem module, wherein the storage device manages virtual volumes that can be addressed by commands directed to a controller of the storage device; and

wherein the filesystem module is configured to associate files in the filesystem with the virtual volumes, respectively, and wherein the filesystem module is further configured to enable the content of the files to be stored in the virtual volumes while managing the files as filesystem objects.

9. A computing device according to claim **8**, wherein the filesystem volume comprises metadata indicating which of the virtual volumes are associated with which of the files.

10. A computing device according to claim **8**, wherein the file system module is configured to handle one or more types of operations directed to arbitrary files of the filesystem by determining, from metadata content of the arbitrary files and/or from filesystem metadata, whether the arbitrary files are associated with virtual volumes, the determining controlling how and/or whether the one or more types of operations are performed by the filesystem.

11. A computing device according to claim **8**, wherein the filesystem module comprises a filter configured to intercept filesystem operations directed to content of the files that are associated with the virtual volumes.

12. A computing device according to claim **11**, wherein the filesystem is configured to either prevent updates to the content of the files via the filesystem or redirect the updates to the virtual volume.